

Massachusetts Institute of Technology
Instrumentation Laboratory
Cambridge, Massachusetts

LUMINARY Memo #27, Rev. 1

TO: Distribution
FROM: Allan Klumpp
DATE: September 26, 1968
SUBJECT: FINDCDUW - Guidance Autopilot Interface Routine

COVER SHEET

This revision is changed from the original issue as follows:

- 1) The provisions for operation of FINDCDUW in other than the normal PNGCS-AUTO mode with the engine on are incorporated. This affects pages 6, 8, and 9 of the text and pages 2, 3, and 6 of Fig. 2.
- 2) The gimbal angles required by guidance are shown delivered to Noun 22. This affects only page 12 of the text, where CDUC is shown stored in the LGC registers of Noun 22, CPHI, CTHETA, CPSI.
- 3) A table of aliases is added (page 12 & 13) relating names used herein to names of corresponding LGC registers.
- 4) Figure 3 is added to clarify the process of erection of the commanded vehicle frame.

I. General Information

FINDCDUW is the interface routine between the various powered flite guidance programs and the digital autopilot. (The "W" in FINDCDUW stands for window control, and distinguishes the present routine from the old FINDCDUD Routine which was never used in the window control mode and still exists, along with an early version of FINDCDUW, in Program SUNDANCE.) FINDCDUW receives from the guidance programs thrust and window pointing commands and provides the autopilot with gimbal angle increments, commanded attitude rates, and commanded attitude lag angles (which account for the angles by which the body will lag behind a ramp command in attitude angle, due to the finite angular accelerations available). FINDCDUW alines the estimated thrust vector with the unit thrust command vector, and, when in the automatic X axis control mode, alines the +Z half of the LM ZX plane with the window command vector.

FINDCDUW is designed to the following objectives:

1. To produce the normal, small angle, maneuvers determined by the guidance inputs at essentially constant attitude rate, reaching the terminal attitude at the completion of the two second computation period.
2. To produce the gross maneuvers, which are required at the boundaries between guidance phases or upon abort, with a continuous, rate-limited attitude profile extending over whatever time duration is required to complete the maneuver.

3. To avoid excessive middle gimbal angle in all maneuvers, never reaching the critical middle gimbal angle during a maneuver in which the terminal middle gimbal angle is less in magnitude than the critical value; to display an alarm code when inputs to FINDCDUW determine a terminal middle gimbal angle in excess of the critical value, and to limit the middle gimbal excursion at the critical value.
4. To permit manual X axis control when the flag FLAUTOX is 0, not interfering when this control is being exercised, not drifting about the X axis when manual control is permitted but not exercised.
5. To provide identical interfaces except with alternate rate limit values when the Command and Service Modules are docked.

The concept of FINDCDUW is a departure from previous concepts for performing this interface function which is computationally more direct and automatically produces maneuvers which avoid excessive middle gimbal angle. The concept of FINDCDUW is as follows. The thrust and window pointing commands, and the location of the thrust vector relative to spacecraft axes, determine a required, or terminal, spacecraft attitude. The gimbal angles corresponding to this terminal attitude are computed, and inputs to the autopilot are generated which drive each gimbal angle directly from its initial value to its terminal value.* Provided the attitude is not initially in gimbal lock, and provided the input commands do not yield a terminal attitude in gimbal lock, it is impossible to pass through gimbal lock during the maneuver since the middle gimbal angle is confined to the range between its initial and terminal values. Other schemes for gimbal lock avoidance produce similar maneuvers, but are computationally less direct.

Control of spacecraft attitude by this gimbal-drive process is illustrated in Figure 1 and described for normal operation as follows. Two fictitious coordinate

* Except the outer gimbal angle profile may not be monotonic in the geometrically complex case of a large maneuver about multiple axes at substantial middle gimbal angle and with magnitude limiting of the X axis attitude angle change on at least one pass thru FINDCDUW.

frames are defined. The first, called the current vehicle frame, is computed using the desired gimbal angles, CDUD's, maintained by the digital autopilot. The unit vectors of the current vehicle frame would be identically the body unit vectors if the desired gimbal angles were identically the actual gimbal angles. The thrust direction filter is processed in the current vehicle frame to yield the unit thrust vector in vehicle coordinates \underline{u}_{FV} . The second frame, called the commanded vehicle frame, is computed using \underline{u}_{FV} , the unit thrust command \underline{u}_{FDP} , and the unit window command \underline{u}_{WDP} . Commanded gimbal angles, CDUC's, are computed using the unit vectors of the commanded vehicle frame. The commanded middle gimbal angle is magnitude limited, and an alarm code is displayed if limiting changed the value. Unlimited gimbal angle changes are computed as the differences between the commanded and the desired gimbal angles. The unlimited gimbal angle changes are transformed to yield attitude angle changes, then limited according to whether the CSM is docked. The X axis attitude change is set to zero if X axis override is permitted or if the unit thrust and unit window commands are too nearly parallel or anti-parallel or if the Y axis or Z axis unlimited gimbal angle change exceeds 45° in magnitude. The resulting attitude angle changes are transformed back to yield the gimbal angle changes. The gimbal angle changes are multiplied by the ratio of the autopilot control sample period δT to the computation period ΔT^* , to yield the gimbal angle increments, which constitute the angle interface with the autopilot. The autopilot adds the gimbal angle increments to the CDUD's each control sample period so that (provided none of the preceding limiting changed any value, normally it does not) the CDUD's are driven to equality with the CDUC's during the succeeding computation period. The digital autopilot closes a control loop between the CDUD's and the measured gimbal angles, CDU's, using, in addition to gimbal angle errors which it computes, the commanded attitude rates and lag angles supplied by FINDCDUW.

* Throughout this Luminary memo δ denotes "increments" of the associated variable which are to occur each control sample period of the digital autopilot, 0.1 second, whereas Δ denotes "changes" of the associated variable which are to occur each computation period, 2 seconds.

It is important that in computing the gimbal angle changes, no reference is ever made to the measured gimbal angles, CDU's. If the CDU's were used as the basis of comparison, one could expect autopilot instability due to feeding back attitude motion due to fuel slosh at the computation frequency. FINDCDUW avoids all reference to the measured spacecraft attitude (except in a few exceptional cases, which will be dealt with in the next section) and still produces zero steady state thrust direction error.

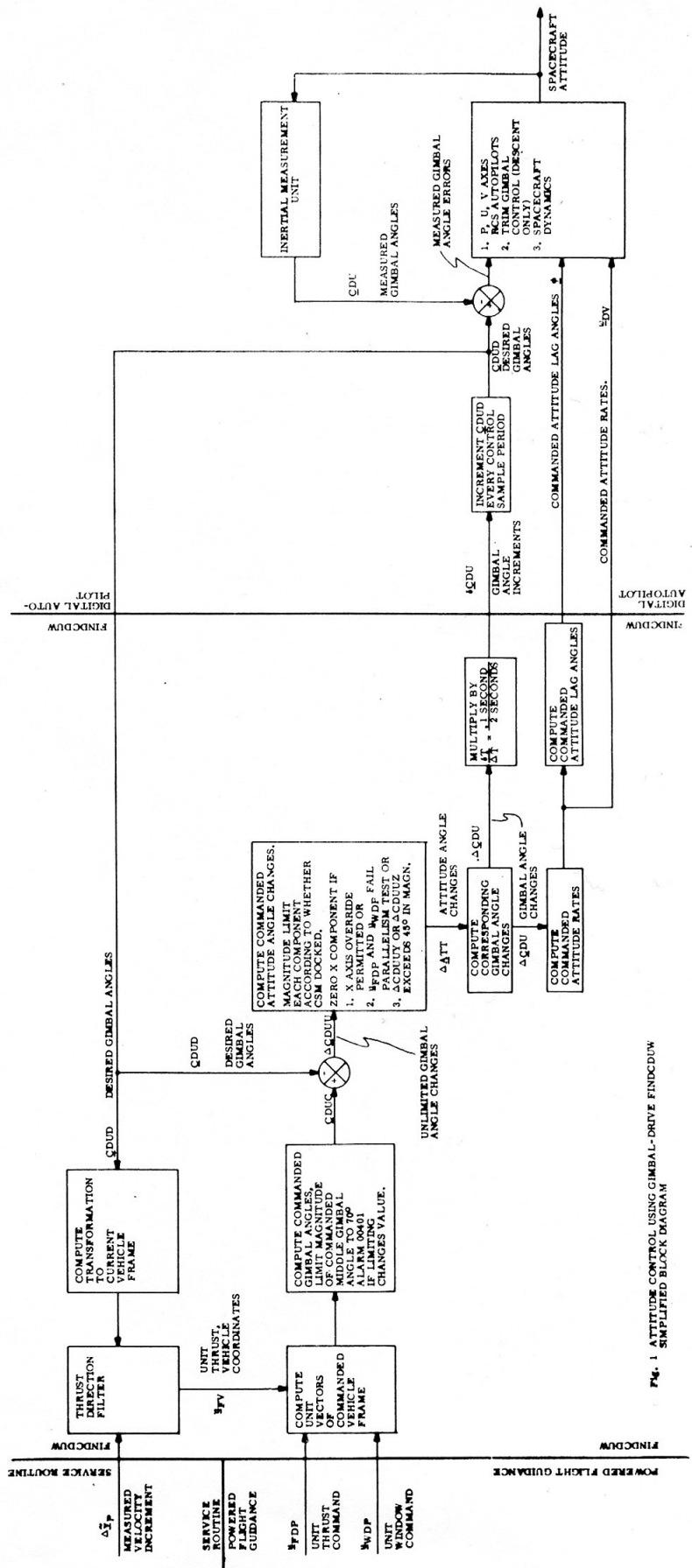


Fig. 1 ATTITUDE CONTROL USING GIMBAL-DRIVE FINDC-DUW SIMPLIFIED BLOCK DIAGRAM

II. Computations

The description of this section follows Figure 2. Prior to engine ignition for each of the major mission modes, the thrust direction filter and the unit window command are initialized. The thrust direction filter must be initialized for each burn because there is not necessarily any relationship between the terminal thrust offset of one burn and the initial thrust offset of the succeeding burn. For computational efficiency, all components of the unit thrust vector in vehicle coordinates, \underline{u}_{FV} , are initialized even though the X component u_{FXV} is never used; the same unit X vector is used to initialize the unit window command \underline{u}_{WDP} . The unit window command must be initialized because there is no requirement for this vector to be supplied by guidance when X axis override is allowed, and FINDCDUW takes its unit which could be disastrous if its contents were random.

The initialization which must be done each pass thru FINDCDUW amounts to acquisition and preparation of constants and input data. These are:

1. θ_{XM} , θ_{YM} , θ_{ZM} , the maximum angular changes, are selected depending upon whether or not the Command and Service Modules are docked.
2. FLAGGOODW is initialized to 0 or 1 according to whether X axis override is permitted or inhibited. This flag indicates whether the unit window command is good for control about the X axis.
3. OVFINDD, the interpretive overflow indicator, is set to zero.
4. If the switches on the LM console indicate PNGCS control and AUTO mode, flag PNGCS-AUTO-NOT is set to zero and the CDUD's are fetched for the thrust direction filter; otherwise flag PNGCS-AUTO-NOT is set to 1 and the CDU's are fetched. Fetching the CDU's for the thrust direction filter when not in PNGCS-AUTO maintains the thrust direction filter even when the autopilot does not maintain the CDUD's, and produces no autopilot instability because no autopilot commands are issued.
5. \underline{u}_{XVDP} and \underline{u}_{ZVDP} , the X and Z axes of the commanded vehicle frame, are initialized to the unit thrust command \underline{u}_{FDP} and the unit window command \underline{u}_{WDP} . If unitizing either of these input command vectors

fails either because a vector is too short or too long, the inputs are presumed unsatisfactory for attitude control, alarm 00402 is issued, attitude motion is stopped, and FINDCDUW returns to the calling program.

6. $\tilde{\Delta \underline{V}_P}$, the velocity increment vector measured by the accelerometers over the previous computation period, is unitized for the thrust direction filter. If this produces overflow, the thrust direction filter is bypassed.

As the first step in the thrust direction filter, $\tilde{\underline{u}_{FP}}$, the unit measured thrust in platform coordinates, is transformed to current vehicle coordinates to yield the unit measured thrust $\tilde{\underline{u}_{FV}}$. Then each of the two components, Y and Z, of the old unit thrust vector is subtracted from the corresponding component of the unit measured thrust, multiplied by the filter gain, and limited in magnitude to the maximum change which can be expected in one computation period, which yields the thrust direction changes Δu_{FYV} and Δu_{FZV} . The maximum changes are based on the slew rate of the trim gimbal. Each of these thrust direction changes is then added to the corresponding component of the old unit thrust vector and limited in magnitude to the maximum thrust offset which can be expected, yielding the new unit thrust vector \underline{u}_{FV} . The maximum thrust offset is based on the maximum displacement of the trim gimbal, the elastic deflection of the gimbal mount, the mounting alignment error, and the deflection of the thrust vector relative to the nozzle centerline. This completes the thrust direction filter, it is not necessary to take the unit of the resulting vector.

In order to compute the axes of the commanded vehicle frame, it is first necessary to select a suitable unit window command. The unit window command obtained from guidance may not be suitable because 1) X axis override may be permitted or 2) it may fail the test PARLTEST (a test for parallelism of anti-parallelism to the unit thrust command). If the unit window command cannot be used for either reason, it is necessary to fetch an alternate such that the resulting axes of the commanded vehicle frame will provide zero steady state thrust pointing error. Any vector lying in the +Z half of the IM symmetry plane will satisfy this requirement.

Among these, the vector least likely to be parallel or anti-parallel to the unit thrust command is the Z axis. However, it is possible during a large maneuver that the Z axis also fails PARLTEST, and in this case a third alternative, the -X axis, is selected. Because the Z axis and the -X axis cannot both be parallel to the unit thrust command, PARLTEST is omitted for -X. The Z and -X axes selected are of the body and not the vehicle frame, simply to save program and time, as the service routine provides the body axes for all users. Using a body axis for window pointing causes no stability problem because any time a body axis is used, FLAGGOODW is set to zero so that the attitude angle change about the X axis will also be set to zero. Thus actual spacecraft motion about the X axis produces through FINCDUW no attitude increment command about the X axis and greatly diminished commands about the Y and Z axes.

Figure 3 illustrates the erection of the commanded vehicle axes. The components of the unit thrust vector in vehicle coordinates, u_{FYV} and u_{FZV} , are treated as small angles, and the corresponding approximations are made. The commanded vehicle axes are computed in two steps. First an orthogonal set is erected with the X axis along the unit thrust command and the Y axis normal to the plane of the unit window command and the unit thrust command. Then the X axis is displaced relative to this initial frame according to the thrust offset. The final Y axis is oriented normal to the initial Z axis and the final X axis, and final Z axis is computed to complete the right hand orthogonal triad. With the coordinate frame erected this way, there is a window pointing error; that is, the unit window command lies out of the ZX plane by approximately the thrust offset angle about the Z axis multiplied by the sine of the angle between the Z axis and the unit window command. MIT simulations indicate that the thrust offset angle about the Z axis during the approach phase of the lunar landing is about 3/4 to 1 degree.

The commanded gimbal angles, CDUC's, are derived from the matrix just computed, whose row vectors are the unit axes of the commanded vehicle frame expressed in platform coordinates. This matrix may be derived in terms of the sines and cosines of the gimbal angles by starting with the matrix whose rows are these unit vectors expressed in commanded vehicle coordinates

(the identity matrix) and rotating the row vectors through the negative of the outer, middle, and inner gimbal angles in that order. The resulting matrix may be used to identify the elements of interest of the matrix of the preceding paragraph as

$$\begin{aligned}c_0 &= \cos(\text{CDUCZ}) \cos(\text{CDUCY}) \\c_1 &= \sin(\text{CDUCZ}) \\c_2 &= -\cos(\text{CDUCZ}) \sin(\text{CDUCY}) \\c_4 &= \cos(\text{CDUCZ}) \cos(\text{CDUCX}) \\c_7 &= \cos(\text{CDUCZ}) \sin(\text{CDUCX})\end{aligned}$$

The subroutine NB2CDUSP extracts the commanded gimbal angles, CDUC, by exploiting these relationships. $|\text{CDUCZ}|$ is $< 90^\circ$, even before limiting.

If flag PNGCS-AUTO-NOT is set or if the engine is off, a STOPRATE is issued and control is returned to the caller of FINDCDUW.

The unlimited gimbal angle changes are computed by modular subtracting the autopilot's CDUD's from the CDUC's. The modular subtractions yield the smallest angular differences, e.g. $177^\circ - (-178^\circ) = -5^\circ$, not $+355^\circ$. If either the Y axis or the Z axis unlimited gimbal angle change exceeds 45° in magnitude, FLAGGOODW is set to zero. This is to prevent false starts in attitude about the X axis, as will be explained in the next paragraph, and analyzed in the appendix.

Attitude angle changes, ΔATT , are handled in a vehicle frame not previously described. The X axis of the frame is identical to the X axis of the current vehicle frame, the Z axis is along the middle gimbal axis, and the Y axis completes the orthogonal right hand triad. Thus this frame is the current vehicle frame rotated about the X axis to bring the Z axis into alignment with the middle gimbal axis. This frame is chosen for computational simplicity. The Z axis gimbal angle change ΔCDUZ is identical to the Z axis attitude angle change ΔATTZ . The unlimited Y axis gimbal angle change is multiplied by $\cos(\text{CDUDZ})$ to yield the unlimited Y axis attitude angle change, this is limited and divided by $\cos(\text{CDUDZ})$ to yield the limited Y axis gimbal angle change ΔCDUY . The unlimited X axis attitude angle change is

computed using the unlimited X and Y axes gimbal angle changes. Note that the modular subtraction always produces a result under 180° in magnitude. This computation produces the smallest angle total rotation required about the X axis, although the validity diminishes as the gimbal angle changes become large. Under certain conditions described in the appendix, the X axis attitude angle change computed will be of opposite sign compared to the succeeding pass. To prevent starting such a maneuver in the wrong direction about the X axis, ΔATTX is set to zero whenever either the Y axis or the Z axis unlimited gimbal angle change exceeds a limit value. Finally the limited X axis gimbal angle change, ΔCDUX , is computed using the limited attitude angle change, ΔATTX , and the limited gimbal angle change ΔCDUY .

With all limiting completed, it is now possible to compute a consistent set of input data for the digital autopilot using the ΔCDU 's. The ΔCDU 's are divided by the computation period and the resulting gimbal angle rates are transformed to yield the commanded attitude rates ω_{DV} . The gimbal angle changes are multiplied by the ratio of the control sample period to the computation period to yield the gimbal angle increments δCDU . Finally, the commanded attitude lag angles ϕ , which account for the angles by which the attitude will lag behind a ramp angular command due to the finite angular accelerations available, are computed using the available angular accelerations α (supplied by the autopilot), and then individually magnitude limited. It should be noted that these data fed to the autopilot are consistent only at the time they are computed. During the succeeding computation period the CDUD 's change at constant rate, the attitude is forced to follow the CDUD 's, therefore the actual attitude rate is not constant. The attitude rate will vary considerably in direction if the middle gimbal angle is large in magnitude, and the attitude rate will vary in magnitude during the computation period if the middle gimbal angle changes. The rate and lag angle data supplied by FINDCDUW become increasingly obsolete during the succeeding computation period. As a worst case example, a maneuver produced by a large step change in the inputs to FINDCDUW resulting in a large total attitude

change at a constant middle gimbal angle of 70° would entail an inner gimbal angle change of over 58° in one computation period and a rotation of the attitude rate vector through the same angle about the Y stable member axis. Thus the component of attitude rate normal to the Y stable member axis would be changed almost 100%. With the present FINDCDUW design, on the succeeding pass the rate and lag angle data would be corrected, but with constant middle gimbal angle there would be no change in the gimbal angle increments fed to the autopilot, provided the inputs to FINDCDUW were unchanged and the maneuver was still at least one computation period from completion. Therefore there would be no transient in the actual spacecraft attitude rate.

Input-Output Nomenclature and Table of Aliases

NAME HEREIN

NAME OF LGC REGISTER

Flags

FLDOC	CSMDOCKD	LM docked flag = 0 LM not docked ≠ 0 LM docked
FLAUTOX	XOVINHIB	X axis override = 0 override permitted ≠ 0 override not permitted
FLAGOODW	FLAGOODW	
FLPAUTNO	FLPAUTNO	

Variables

OVFIND	OVFIND	Computer overflow indication
\underline{u}_{FDP}	UNFC/2	Unit vector in the desired thrust direction in Platform coordinate system.
\underline{u}_{WDP}	UNWC/2	Unit vector in the desired window direction in Platform coordinate system.
$\underline{u}_{XVDP}, \underline{u}_{YVDP}, \underline{u}_{ZVDP}$	UNX/2, UNY/2, UNZ/2	
\underline{u}_{FV}	UNFV/2	Filtered unit measured thrust direction vector.
ΔV_P	DELV	PIPA measured velocity change since the last sample time.
\underline{CDU}	CDU	Measured gimbal angles.
\underline{CDUD}	CDUD	The desired gimbal angle as maintained by the autopilot.
\underline{CDUC}	CPHI, CTHETA, CPSI (NOUN 22)	Commanded gimbal angles
$\Delta \underline{CDUU}$	-DELGMB	
$\Delta \underline{CDU}$	-DELGMB	
ΔA_{TTX}	-DELGMB (SCRATCH STORAGE)	
C_{PV} (\underline{CDUV})	3 × 3 matrix which transforms a vector in platform coordinates to the bad coordinates defined by \underline{CDUV} . No LGC equivalent. Transformations done in LGC by subroutine *SMNB*. One Euler angle at a time.	

Variables (Cont.)

\underline{u}_{ZBP}	ZNBPIP
\underline{u}_{XBP}	XNBPIP
$\delta \underline{CDU}$	DELCDUX, DELCDUY, DELCDUZ Commanded incremental CDU change sent to the autopilot to occur in one autopilot sample period.
$\underline{\omega}_{DV}$	OMEGAPD, OMEGARQD, OMEGARD Commanded body attitude rate sent to the autopilot.
ϕ	DELPEROR, DELQEROR, DELREROR
α	1JACC The angular accelerations which can be achieved by the RCS.

Modes

PNGCS MODE	LM console switch yes PNGCS Control no
AUTO MODE	LM console switch yes PNGCS Autopilot Control no

**INITIALIZATION
PRE-ENGAGEMENT**

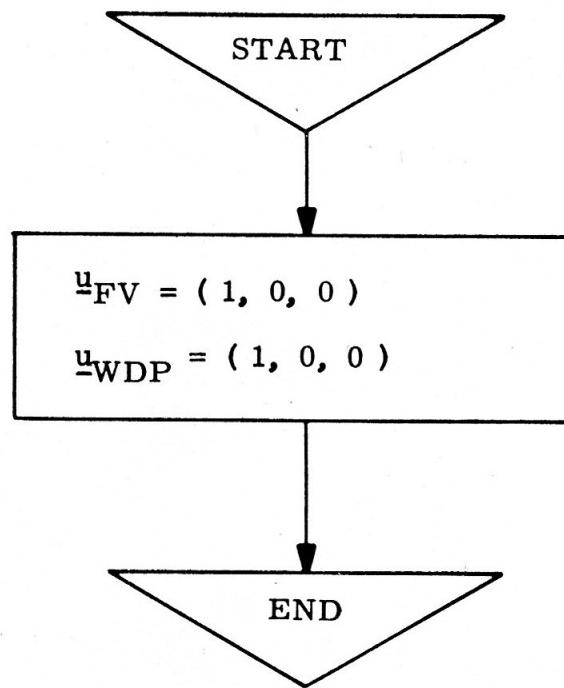
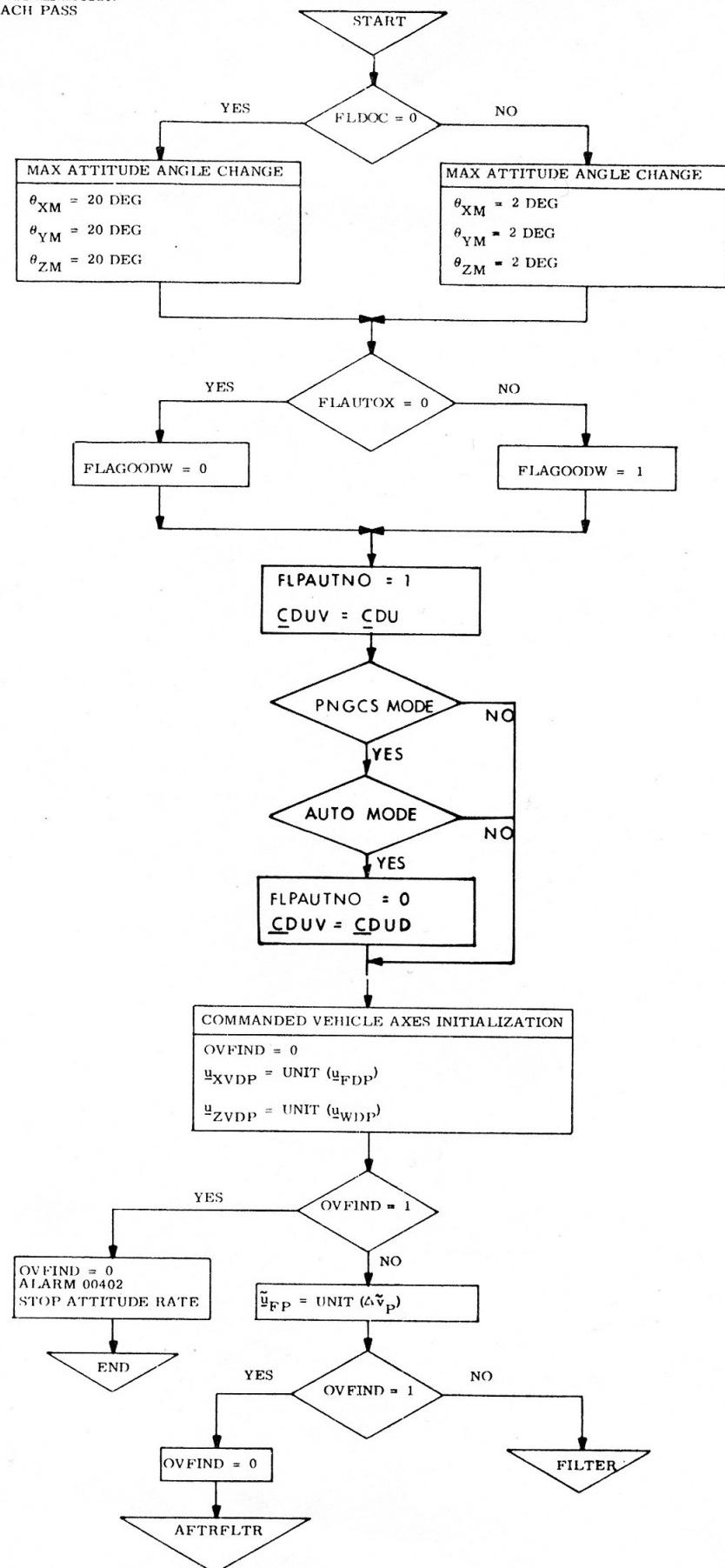


Fig. 2 (p 1)

INITIALIZATION
EACH PASS



THRUST DIRECTION
FILTER

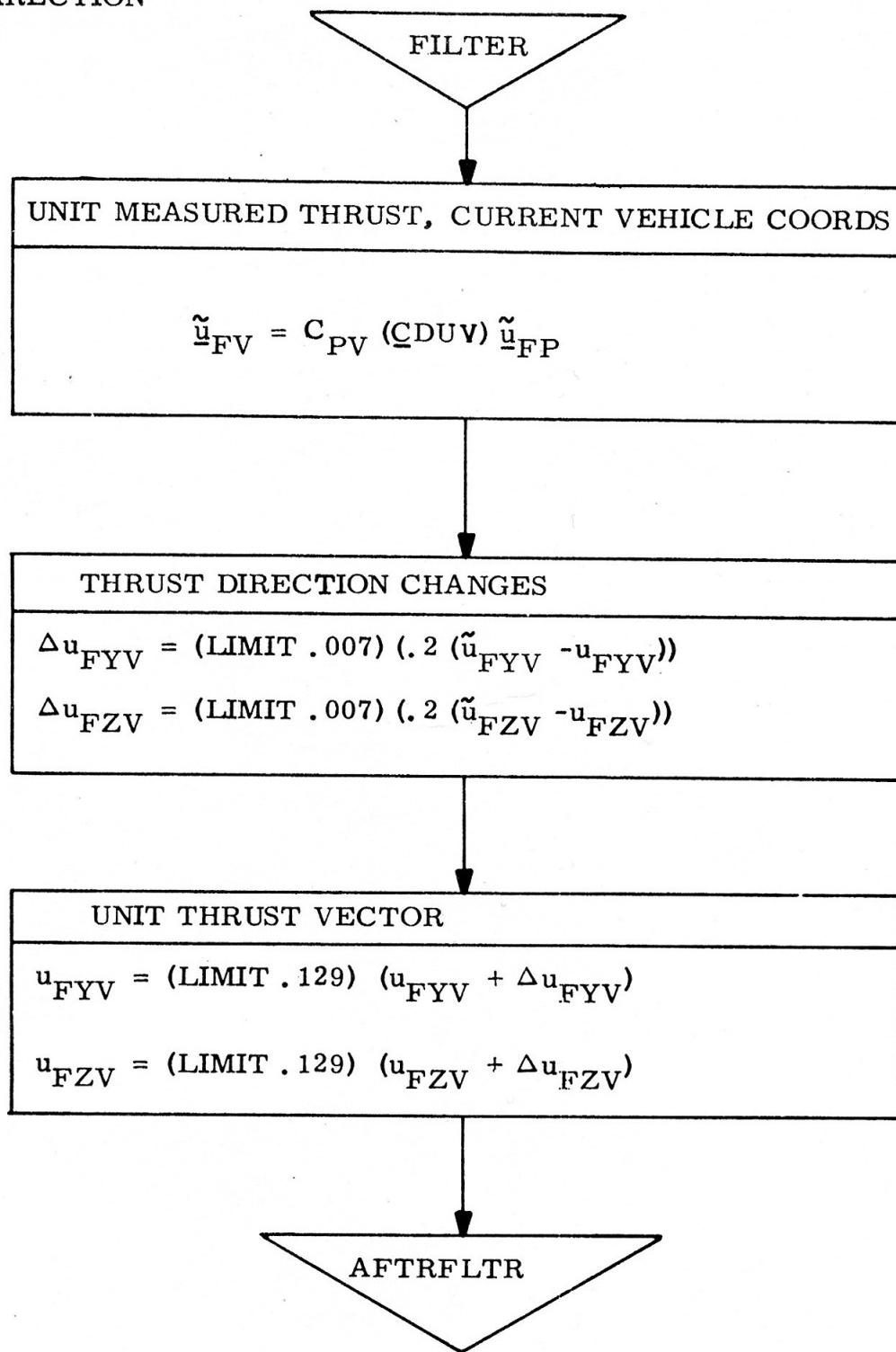


Fig. 2 (p 3)

UNIT WINDOW COMMAND
SELECTION

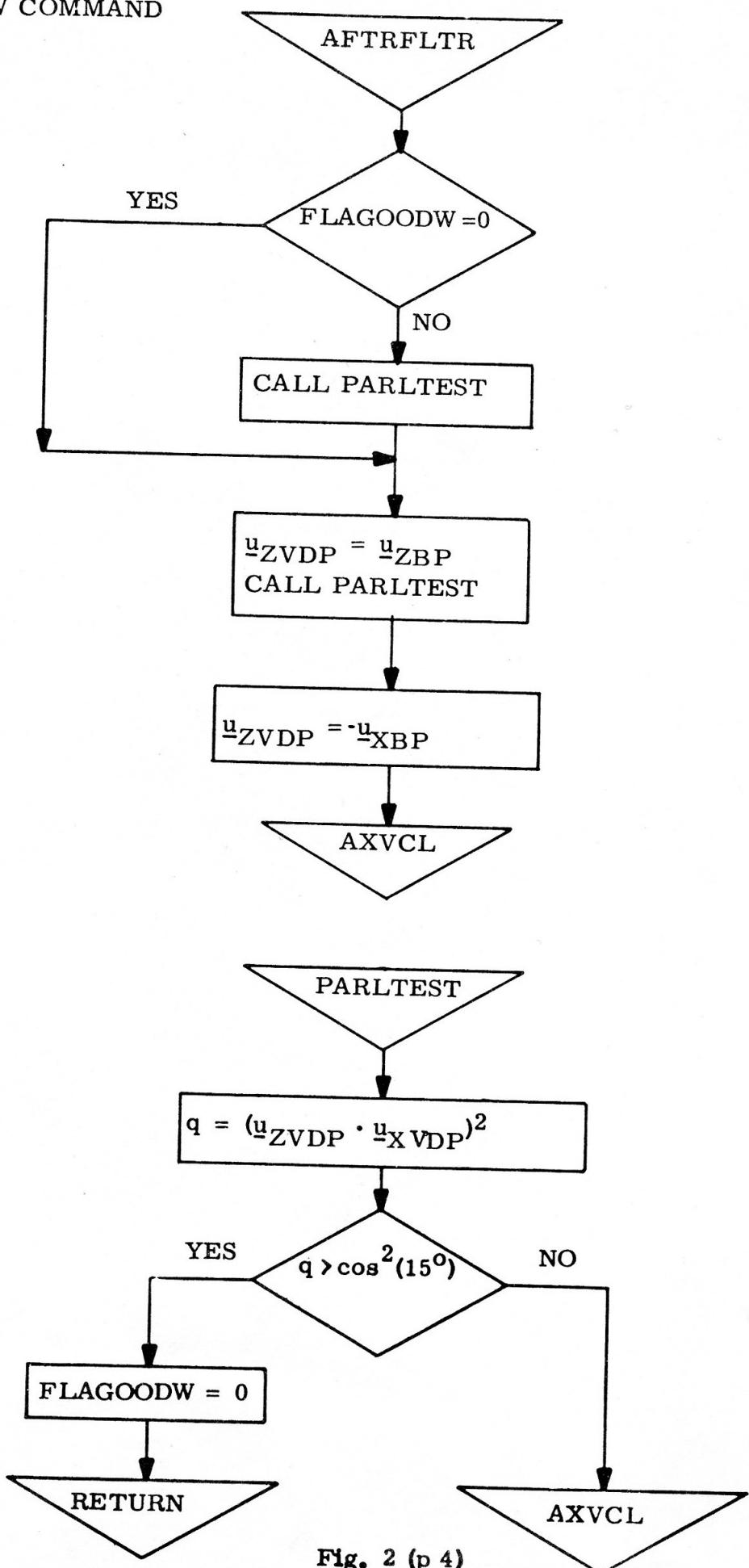
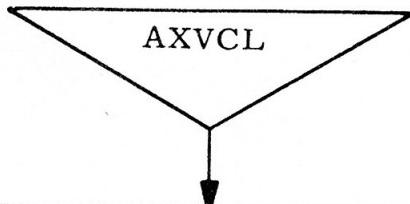


Fig. 2 (p 4)

COMMAND
VEHICLE AXES



COMMAND VEHICLE AXES, INITIAL ITERATION

$$\underline{u}_{YVDP} = \text{UNIT } (\underline{u}_{ZVDP} \times \underline{u}_{XVDP})$$

$$\underline{u}_{ZVDP} = \underline{u}_{XVDP} \times \underline{u}_{YVDP}$$

X AXIS CORRECTION FOR THRUST OFFSET

$$\underline{u}_{XVDP} = \text{UNIT } (\underline{u}_{XVDP} - u_{FYV} \underline{u}_{YVDP} - u_{FZV} \underline{u}_{ZVDP})$$

COMMAND VEHICLE AXES, FINAL ITERATION

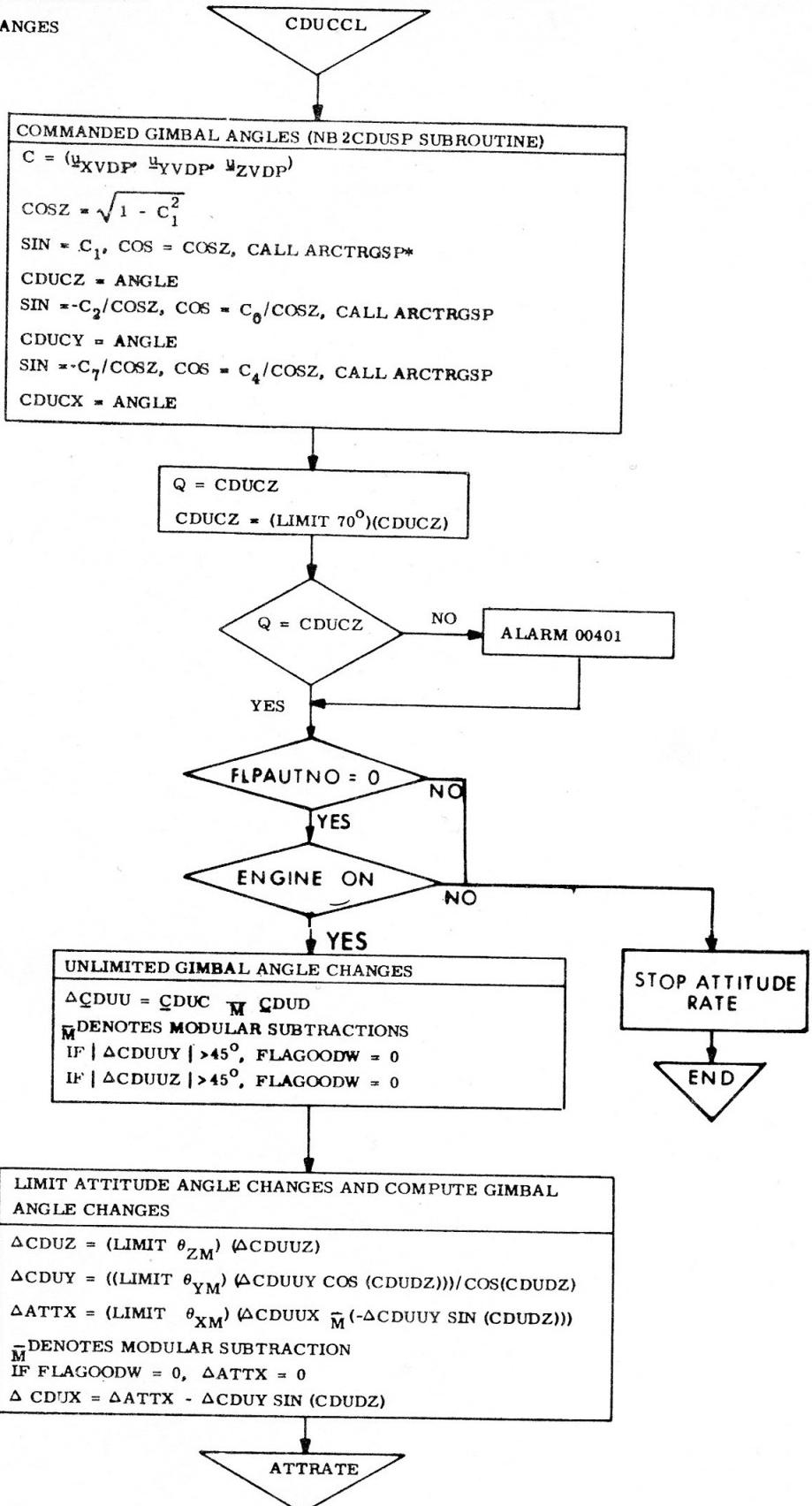
$$\underline{u}_{YVDP} = \underline{u}_{ZVDP} \times \underline{u}_{XVDP}$$

$$\underline{u}_{ZVDP} = \underline{u}_{XVDP} \times \underline{u}_{YVDP}$$



Fig. 2 (p 5)

COMMANDED GIMBAL ANGLES AND
GIMBAL ANGLE CHANGES



*ARCTRGSP IS A SUBROUTINE WHICH COMPUTES AN UNAMBIGUOUS ANGLE ANYWHERE IN THE CIRCLE, GIVEN THE SINE AND THE COSINE.

AUTOPilot DATA

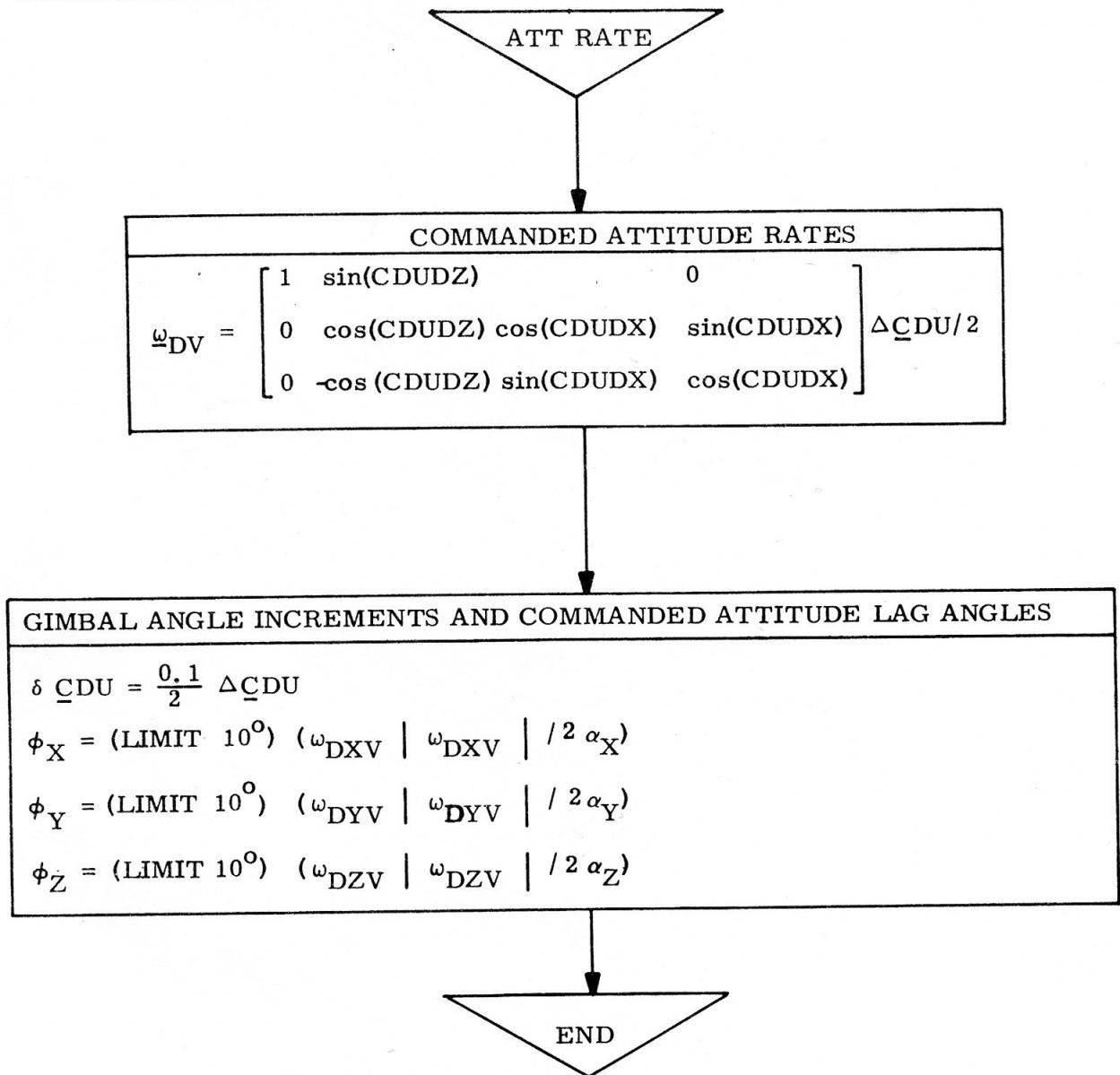
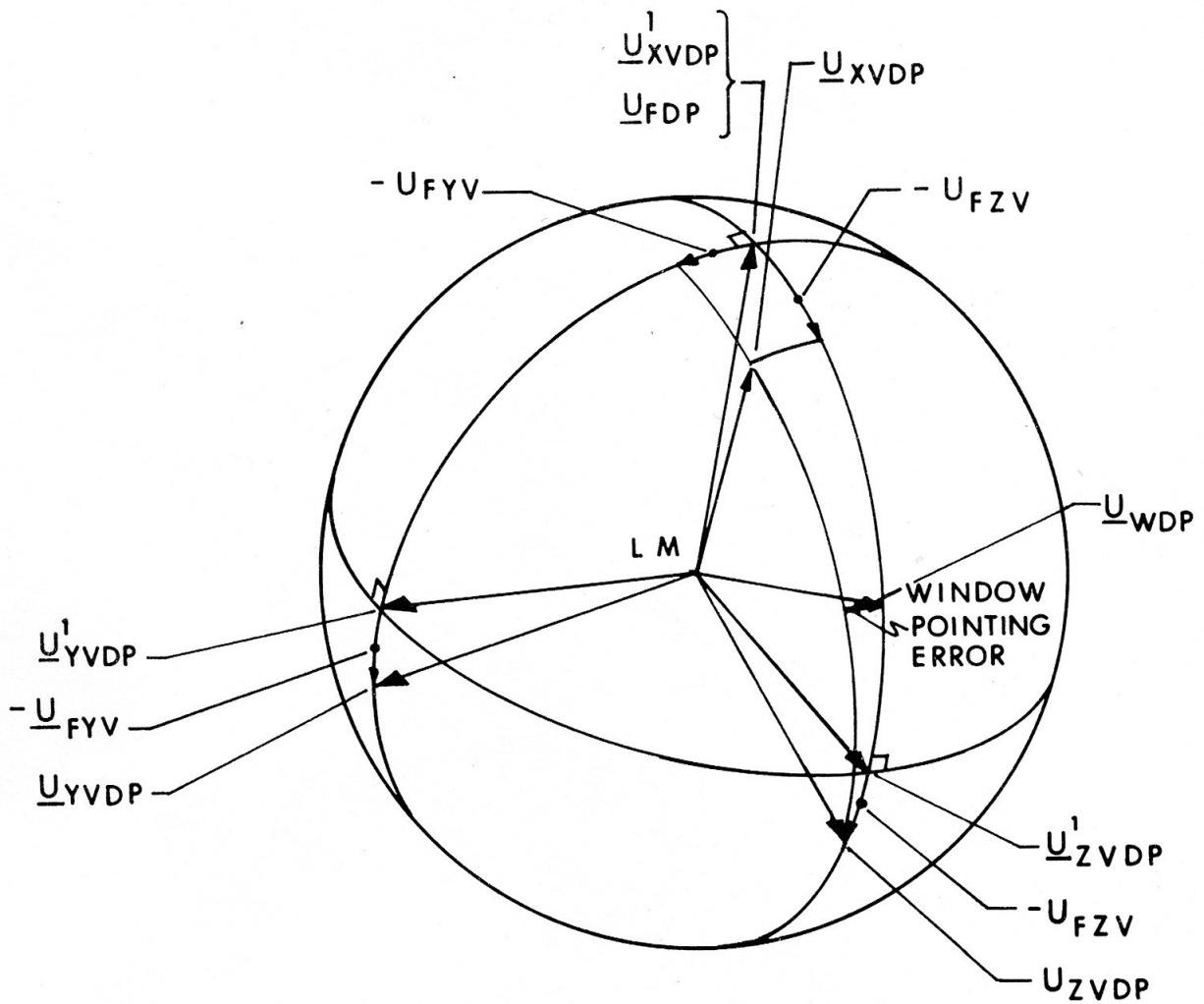


Fig. 2 (p 7)



NOTE: The vehicle coordinate axes are computed in two steps. Vectors computed the first step are identified by the superscript 1. Final values have no superscripts.

Fig. 3 Geometry of Erection of Commanded Vehicle Axes Viewed on a LM Centered Unit Sphere.

APPENDIX

False Starts About the X Axis

Without the provisions developed here, it is possible to compute an unlimited attitude angle change about the X axis on one pass, rotate such as to supposedly diminish the value, and find on the succeeding pass that it has in fact increased. If the value on one pass is near 180° on one side, the value on the succeeding pass might lie near 180° on the opposite side, and the spacecraft would be required to reverse direction of rotation about the X axis. The criterion developed here prevents issuance of an X axis attitude angle change command whenever the value computed on the succeeding pass could be of increased magnitude.

The two equations of this paragraph provide insight, but do not yield a satisfying solution because of the time-discrete character of the computations. The unlimited attitude angle change about the X axis is computed as part of the equation for ΔATTX on page 6 of Figure 2. For the purpose of this derivation, the modular subtraction of a minus can be replaced by an addition to yield

$$\Delta\text{ATTUX} = \Delta\text{CDUUX} + \Delta\text{CDUY SIN(CDUDZ)}. \quad (1)$$

Note that the unlimited gimbal angle changes are used. The time derivative of the attitude angle change is

$$\frac{d\Delta\text{ATTUX}}{dt} = \frac{d\Delta\text{CDUUX}}{dt} + \frac{d\Delta\text{CDUY}}{dt} \text{ SIN(CDUDZ)} + \Delta\text{CDUY COS(CDUDZ)} \frac{d\text{CDUDZ}}{dt}. \quad (2)$$

The sum of the first two terms on the right is the attitude rate about the X axis. This sum is the quantity controlled, and is ordinarily commanded in the direction to diminish ΔATTUX . The difficulty is that the third term

on the right is not controlled by the X axis command. Therefore it may offset and overcome what is done in controlling the first two terms.

The solution comes from considering the X axis unlimited attitude angle change computed on two passes. The value computed may be either positive or negative. To avoid cumbersome nomenclature only the positive case will be treated. For this case we wish a criterion to indicate when the unlimited attitude angle change computed on the second pass will be less than that computed on the first pass, assuming gimbal angle increment commands are issued to the autopilot on the first pass, and assuming the input commands to FINDCDUW are unchanged on the second pass. The unlimited attitude angle changes computed on the two passes are

$$\Delta\text{ATTUX}_1 = \Delta\text{CDUUX}_1 + \Delta\text{CDUU}Y_1 \sin(\text{CDUDZ}_1), \quad (3)$$

$$\Delta\text{ATTUX}_2 = \Delta\text{CDUUX}_2 + \Delta\text{CDUU}Y_2 \sin(\text{CDUDZ}_2). \quad (4)$$

The change in the computed value is, subtracting (3) from (4)

$$\begin{aligned} \Delta\text{ATTUX}_2 - \Delta\text{ATTUX}_1 &= \Delta\text{CDUUX}_2 - \Delta\text{CDUUX}_1 \\ &\quad + \Delta\text{CDUU}Y_2 \sin(\text{CDUDZ}_2) - \Delta\text{CDUU}Y_1 \sin(\text{CDUDZ}_1). \end{aligned} \quad (5)$$

But, with the X axis attitude angle change limited, the equation on page 6 of Figure 2 which produced the limiting may be rewritten as

$$\Delta\text{CDUX}_1 + \Delta\text{CDUY}_1 \sin(\text{CDUDZ}_1) = \theta_{XM}. \quad (6)$$

This is a relationship for the changes which will actually be made to CDUDX and CDUDZ by the digital autopilot during the interval between the first and second passes thru FINDCDUW. With the commands to FINDCDUW unchanged on the second pass, the unlimited gimbal angle changes computed on the second pass will be decremented by the same amounts. That is

$$\Delta CDUUX_2 = \Delta CDUUX_1 - \Delta CDUX_1, \quad (7)$$

$$\Delta CDUUY_2 = \Delta CDUUY_1 - \Delta CDUY_1. \quad (8)$$

Solving Equations (7) and (8) for $\Delta CDUX_1$ and $\Delta CDUY_1$, substituting into Equation (6), and rearranging terms yields

$$0 = \Delta CDUUX_2 - \Delta CDUUX_1 + (\Delta CDUUY_2 - \Delta CDUUY_1) \sin(CDUDZ_1) + \theta_{XM}. \quad (9)$$

Subtracting (9) from (5) yields

$$\Delta ATTUX_2 - \Delta ATTUX_1 = \Delta CDUUY_2 [\sin(CDUDZ_2) - \sin(CDUDZ_1)] - \theta_{XM}, \quad (10)$$

which we wish to be negative. Taking the case when $\Delta CDUUY_2$ is positive (the negative case yields the same criterion for $|\Delta CDUUY_2|$), if $CDUDZ_2 < CDUDZ_1$, the right side is always negative, but if $CDUDZ_2 > CDUDZ_1$ the criterion is

$$\Delta CDUUY_2 < \frac{\theta_{XM}}{\sin(CDUDZ_2) - \sin(CDUDZ_1)}. \quad (11)$$

Because the change in sines is always less than the change in angles, it is conservative to replace the sines by the angles. And since Z axis gimbal angles and attitude angles are identical, the worst case is when the change in the middle gimbal angle is the limit value. In this case the criterion becomes

$$|\Delta CDUUY_2| < \frac{\theta_{XM}}{\theta_{ZM}}. \quad (12)$$

Taking the absolute value makes the criterion applicable to the cases involving other signs of the variables. With $\theta_{XM} = \theta_{ZM}$ the criterion

becomes finally

$$|\Delta CDUUY_2| < 1 \text{ radian} . \quad (13)$$

Because only $\Delta CDUUY_1$ is available at the time the test is made, it is used instead, which produces a more conservative test. Thus if $\Delta CDUUY_1$ is too large in magnitude, the X axis attitude angle change is set to zero, and the X axis attitude profile is thereby rendered essentially monotonic.